On the Attacker's Knowledge in Shared-Key Cryptosystems

Fabrizio Biondi, Thomas Given-Wilson, and Axel Legay

Inria
Email: {fabrizio.biondi,thomas.given-wilson,axel.legay}@inria.fr

Abstract. Recent work has presented max-equivocation as a measure of the resistance of a cryptosystem to attacks when the attacker is aware of the encoder function and message distribution. Here we consider the vulnerability of a cryptosystem in the one-try attack scenario when the attacker has incomplete information about the encoder function and message distribution. We show that encoder functions alone yield information to the attacker, and combined with inferable information about the ciphertexts, information about the message distribution can be discovered. We show that the whole encoder function need not be fixed or shared a priori for an effective cryptosystem, and this can be exploited to increase the equivocation over an a priori shared encoder. Finally we present two algorithms that operate in these scenarios and achieve good equivocation results, ExPad that demonstrates the key concepts, and ShortPad that has less overhead than ExPad.

1 Introduction

The privacy of private communication is a fundamental concern of contemporary computer science. Approaches in this area can be divided into two categories: computation and unconditional security. Computation security relies upon the (assumed) superpolynomial lower bound of computational complexity for some particular functions, e.g. factorization. These lower-bound results are unproven, and are susceptible to advances in algorithmic theory, quantum computing [1], or engineering [2]. Unconditional security is instead based upon information-theoretic results and proven independently of computational hardness or techniques. Thus unconditional security results are both more general and theoretically solid than computational security results.

Initial formal results on unconditional security began with Shannon [3] using key concepts from the formal theory of communication [4]. Shannon considered the transmission of an encrypted message on a public channel between a sender and a receiver that share a cryptographic key. The results formalized that to achieve *perfect secrecy*, where the attacker gains no information about the message, the cryptographic key space must be as large as the message space. Further, the keys must be uniformly distributed over the key space, and the key can be used only once.

In [5] *max-equivocation* was presented as a measure of unconditional security that generalizes perfect secrecy. Max-equivocation measures the resistance of a cryptosystem to attacks from an unconditional attacker, in particular the bounds that can be achieved with a key space smaller than the message space, and also scenarios where the key space is not uniformly distributed. This is done over encoder functions for shared-key cryptosystems where the encoder function and message distribution are known to the attacker. Here we consider the implications of these assumptions, in particular when relaxing them. However, we will always consider that the attacker has the same information that the receiver has, except for the value of the key to avoid our results being reliant on security through obscurity.

When the attacker does not know the actual distribution of messages generated then alternative approaches are required to compute the vulnerability of the cryptosystem. To this end we define *relative vulnerability* that is able to quantify the vulnerability when the attacker's distribution over the messages does not match that of the message *generation*. It turns out that when considering one-try attacks (min-entropy [6]) that the vulnerability of the system can only decrease when the attacker does not know the generator's distribution. That is, an attacker using the wrong distribution can never do better than knowing the generators distribution.

Another path of attack is to consider the information yielded by an encoder function alone. It turns out that knowing an encoder function can yield significant information to an attacker. To formalise this we present a representation of an encoder function as a matrix. The rank of this matrix can be exploited to discover information about the message distribution. In particular, if the rank is maximal then given the distribution over the ciphertexts it is straightforward to determine the message distribution. Conversely, for each rank short of maximal (or each degree of freedom) there is an infinitude of possible message distributions.

It is then natural to consider the attacker's ability to determine the generated message distribution by observing ciphertexts and exploiting the encoder function. We show that determining the ciphertext frequency is straighforward, and that this can be combined with knowledge of the encoder function to converge on some reasonable message distribution. Additionally when the encoder function's matrix representation has maximal rank then the attacker can uniquely determine the generators message distribution. Even when the encoder function's matrix representation does not have maximal rank, the attacker can refine their knowledge of the message distribution via these techniques to converge on a distribution closer to that of the generator. These results make it clear that merely maximising equivocation is not sufficient when the attacker does not already know the generator's message distribution. Indeed it may be better to reduce equivocation slightly and so introduce uncertainty for the attacker over the message distribution.

Building on this knowledge, we consider the scenario where the sender may wish to transmit only the minimal information required to decode the message.

Here it is sufficient for the sender to transmit only a small part of an encoder (or decoder) function that will allow the receiver to determine the message. To this end we introduce a *curried decoder function* that is the minimal amount of the encoder function required to decode the message. Interestingly, this does not require the fixing of an encoder function a priori, and so the sender can craft an encoder function to yield good information-theoretic results for the particular message to be sent. However, a naive approach to this can lead to leaking information to the attacker, and so the choice of encoder function must be made with some care.

We present the ExPad algorithm that can achieve excellent equivocation and good overall leakage properties by considering the above results. This is achieved by combining knowledge of the generator's message distribution with some randomness that selects a sub-set of the messages with similar probabilities. A curried decoder function is then sent that contains these messages as its image and with the key mapping to the original message. We formalize that ExPad has good leakage properties and is reasonable to implement. However, ExPad requires sending an amount of information linear in the size of the key space.

To improve upon this we present the ShortPad algorithm that reduces the information sent to be logarithmic in the size of the key space. This is achieved by exploiting an operation that treats the elements of the message space as a group. A pad is chosen that is a sub-set of the message space, and then the bit string representation of the key is used to choose a subset of this pad to be combined with the operation. The choice of pad can be made to have elements with similar probability in the image, yielding similar equivocation results to ExPad. Further, by exploiting the bit string representation of the key and the pad the transmission is logarithmic in the size of the key space rather than linear. Indeed, ShortPad can also be used with pre-agreed pads and operations to further reduce the required transmission from sender to receiver.

Main Contributions This paper contributes the following main results.

- We introduce *relative vulnerability*, a distance measure between probability distributions that corresponds to the difference in min-entropy leakage if the attacker assumes a message distribution different from the real message distribution.
- We prove that using an incorrect message distribution is never beneficial for the attacker in one-try attack scenarios.
- We prove that the information inferred by the attacker on the message distribution depends on the rank of the matrix representation of the encoder function.
- We show how the attacker can infer the ciphertext distribution and use it to gain information about the message distribution.
- We analyze the equivocation in the scenario where the sender crafts and transmits curried decoder functions optimized for particular messages, thus avoiding revealing the full encoder function, and quantify the security of this approach.

– We present the ExPad algorithm to securely construct and transmit curried decoder functions and prove its effectiveness.
– We present the ShortPad algorithm achieving similar security results to ExPad while transmitting logarithmic rather than linear amounts of information. We show that ShortPad can reduce this further to be a constant albeit with worse leakage results.

Related Work Information-theoretic analysis of shared-key cryptosystems was introduced by Shannon [4, 3], including the concepts of entropy, equivocation, and perfect secrecy. Max-equivocation was introduced by Biondi et al. [5, 7] considering only the case in which the encoder and message distribution are known to the attacker. This work extends the same analysis to the case where the attacker does not possess this knowledge.

Min-entropy was introduced by Smith [6] as an information-theoretic measure of information leakage for one-try attack scenarios. Note that min-entropy leakage is a measure of secrecy, i.e. difference between the attacker's knowledge before and after observation of the system's output, while equivocation is a measure of the resistance of the system after the observation.

Min-entropy leakage is based on vulnerability, i.e. the probability of success of a one-try attack and the complement of Bayes risk. Min-entropy leakage has been generalized to arbitrary gain functions as g-leakage [8], and its induced ordering on channels has been extensively studied [9]. Strong and weak concepts of perfect secrecy based only on vulnerability have been studied [10]. Additive measures of information leakage have been proposed for the one-try attack scenario [11, 12].

Clarkson et al. [13] model the accuracy of the attacker's information about the message as the relative entropy from the attacker's distribution to the real message distribution, and Hamadou et al. [14] follow the same principle to derive bounds on the vulnerability of a system against an attacker with incorrect information. Contrarily to their approach, we compute a direct measure of the vulnerability of the system against an attacker with incorrect information.

Structure of the Paper The structure of the paper is as follows. Section 2 recalls background material. Section 3 presents relative vulnerability and its relation to min-entropy leakage. Section 4 discusses how the message distribution can be inferred from the encoder function by exploiting a matrix representation and considering the rank. Section 5 explains how the attacker can infer the ciphertext distribution and use it to gain information about the message distribution. Section 6 considers transmitting only a curried decoder function to the receiver to hide the full encoder function, and quantifies the information leakage to the attacker. Section 7 presents ExPad, an effective transmission algorithm based on the results of this paper. Section 8 presents ShortPad, an algorithm almost as secure as ExPad transmitting only a logarithmic amount of data compared to ExPad. Section 9 concludes and considers future work.

2 Background

We recall standard definitions and concepts that will be used throughout this work.

The size of a set $\mathcal{S}$ is denoted as $|\mathcal{S}|$. A function $f : \mathcal{A} \to \mathcal{B}$ is *injective* iff $\forall a_1, a_2 \in \mathcal{A}.\ f(a_1) = f(a_2) \Rightarrow a_1 = a_2$.

Basic concepts from probability and information theory can be found in the literature [15], including the definitions of support set $\mathcal{X}$, probability $P(E)$ of an event $E \subseteq \mathcal{X}$, random variable X on $\mathcal{X}$, entropy $H(X)$ of a random variable, and so on. We will write $\rho_{\mathcal{X}}(X)$ for a probability distribution on the random variable X on the support set $\mathcal{X}$, abbreviated to $\rho(X)$ when the support set is unambiguous.

2.1 Shared-Key Cryptosystems

We recall the definition of a shared-key cryptosystem, components, and typical use as the basis for the results and scenarios in this paper. A shared-key cryptosystem can be defined by the following components [7].

Definition 1. *A* (shared-key) cryptosystem *is a 3-tuple* $(\mathcal{M}, \mathcal{K}, \mathrm{enc})$ *where:*

- *the* message space $\mathcal{M}$ *is a finite set of possible messages;*
- *the* key space $\mathcal{K}$ *is a finite set of possible keys;*
- *the* encoder enc *is a function* $\mathcal{M} \times \mathcal{K} \to \mathcal{C}$ *to some space* $\mathcal{C}$ *such that* $\forall k \in \mathcal{K}.\ \mathrm{enc}(\cdot, k)$ *is injective.*

A shared-key cryptosystem induces a ciphertext space $\mathcal{C} = enc[\mathcal{M}, \mathcal{K}]$ as the image of its encoder function and a decoder function as a function $dec : \mathcal{C} \times \mathcal{K} \to \mathcal{M}$ such that $dec(enc(m, k), k) = m$. The existence and uniqueness of such a decoder function is ensured by the requirement that $enc(\cdot, k)$ is injective.

The channel model of a cryptosystem was introduced by Shannon [3]. In this model, the sender wants to send a message $m \in \mathcal{M}$ to the receiver on a public channel that is eavesdropped by an attacker. Initially, the sender and receiver share a secret *key* $k \in \mathcal{K}$.

The sender encodes the message m with key k into the ciphertext c as $c = enc(m, k)$. The sender then sends c to the receiver via a public channel, where c is also eavesdropped by the attacker. Knowing the key k, the receiver decodes the message $m = dec(c, k)$ using the decoder function. Not knowing the key k, the attacker tries to infer m from c. The computational power available to the sender, receiver and attacker is assumed to be unlimited.

The attacker's knowledge about the elements of the communication is modeled by random variables. Let M (resp. K, C) be a random variable on the support set $\mathcal{M}$ (resp. $\mathcal{K}$, $\mathcal{C}$) representing the value of the message m (resp. key k, ciphertext c) according to the attacker.

2.2 Information-Theoretic Analysis

We recall min-entropy leakage as a measure of the vulnerability of a system. Min-entropy leakage was proposed by Smith [6] to quantify the vulnerability of a system against an attacker that, after observing the system's output, has one attempt to guess the value of the systems' secret, this is known as a *one-try attack* scenario. In a shared-key cryptosystem scenario the secret is the message m and the observable is the ciphertext c.

Let M be a random variable representing a secret value and ρ the probability distribution on its support $\mathcal{M}$ modeling the attacker's a priori information about the message. Let C be the random variable representing the ciphertext observed by the attacker.

Since the attacker has only one attempt to guess the secret message, they will guess the message with the highest probability according to their distribution. When multiple messages have the same highest probability, one of them is chosen uniformly at random.

Then the *vulnerability* of the message is the probability that the attacker's guess will be correct:

$$V(M) = \max_{m \in M} \rho(m)$$

and the *conditional vulnerability* of M after observing C is

$$V(M|C) = \sum_{c \in C} \rho(c) \max_{m \in M} \rho(m|c) \ .$$

It is standard to transform vulnerability into entropy and so be able to quantify the difference in bits. The *prior min-entropy* of the message is

$$H_\infty(M) = -\log V(M)$$

while the *conditional min-entropy* or *min-entropy equivocation* is

$$H_\infty(M|C) = -\log V(M|C)$$

and the *min-entropy leakage* is the difference of the two:

$$\mathcal{L}_\infty(M, C) = H_\infty(M) - H_\infty(M|C) \ .$$

Observe that the conditional min-entropy represents the information after c has been observed.

Min-entropy is designed to capture the change in the probability that the attacker will be able to guess the value of the secret in one attempt.

Since the key is also unknown to the attacker, the key's vulnerability and min-entropy can also be computed. Note that since the message and key are chosen independently, then $H_\infty(M, K) = H_\infty(M) + H_\infty(K)$.

Shannon defined *perfect secrecy* as the highest possible security condition attainable by a cryptosystem [3]. It corresponds to the leakage being zero.

Definition 2 (Perfect Secrecy). *A cryptosystem attains* perfect secrecy *in the one-try attack scenario iff*
$\mathcal{L}_\infty(M, C) = 0$.

Shannon also proved that for a cryptosystem to be perfectly secure it is necessary that the key space is at least as large as the message space, i.e. $|\mathcal{K}| \geq |\mathcal{M}|$, making perfect secrecy hard to achieve in practice. Instead, *max-equivocation* [7] has been proposed as the highest possible security condition that is attainable by any cryptosystem, including when $|\mathcal{K}| < |\mathcal{M}|$:

Definition 3 (Max-Equivocation). *A cryptosystem attains* max-equivocation *in the one-try attack scenario iff* $H_\infty(M|C) = H_\infty(K)$.

2.3 Beta and Dirichlet Distributions

The Beta (resp. Dirichlet) distribution can be used to infer the parameters of a binomial (resp. multinomial) distribution from a number of samplings of the distribution. The Beta (resp. Dirichlet) distribution assigns a random variable to each of the elements in the support of a binomial (resp. multinomial) distribution, and the expected value of this random variable converges to the probability of the element. We exploit this later in the paper by using Dirichlet results to converge upon a distribution based upon observations, e.g. the attacker can converge upon the ciphertext distribution by observations of a sufficient number of ciphertexts.

We present the key notations and components of Beta and Dirichlet distributions, but do not revisit the well known results. For greater detail we refer to the literature [16], and for those who are interested only in the results of this paper the details may be intuitively presented in the results.

A Beta distribution is parameterized over two non-negative parameters α and β.

Definition 4. *The Beta*(α, β) *distribution with parameters* (α, β) *defined on* $x \in [0, 1]$ *has probability density function*

$$f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathrm{B}(\alpha, \beta)}$$

where

$$\mathrm{B}(\alpha, \beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1} dx$$

is the Beta function on two parameters.

To generalize the two-parameter Beta distribution to the multi-parameter Dirichlet distribution, we need to introduce a simplex for the probability space over possible multinomial distributions.

Definition 5. *Let a* $(q-1)$*-dimensional* probability simplex Δ_q *be the surface in* $\mathbb{R}^q$ *whose* q *components* $\{x_1, \ldots, x_q\}$ *are greater than zero and sum up to one:*

$$\Delta_q = \left\{ x \in \mathbb{R}^q \,|\, 1 = \sum_{i=1}^{q} x_i \wedge x_i \geq 0 \right\} \quad .$$

A Dirichlet distribution is a probability distribution over the probability simplex Δ_q. The distribution is parameterized over q parameters $\{\alpha_1, \ldots, \alpha_q\}$.

Definition 6. *Let $X = [X_1, \ldots, X_q]$ be a vector of q random variables such that $X_q = 1 - \sum_{i=1}^{q-1} X_i$ and $X_i \geq 0$ for $i = 1, \ldots, q$. Then X has a Dirichlet distribution with parameters $\{\alpha_1, \ldots, \alpha_q\}$ if the probability mass function of X is*

$$X \sim Dir(\alpha) = \frac{1}{\mathrm{B}(\alpha)} \prod_{i=1}^{q} x_i^{\alpha_i - 1}$$

where

$$\mathrm{B}(\alpha) = \frac{\prod_{i=1}^{q} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{q} \alpha_i\right)}, \qquad \alpha = (\alpha_1, \cdots, \alpha_q)$$

and Γ is the gamma function.

Let $\alpha_0 = \sum_{i=1}^{q} \alpha_i$. Then the marginal distribution on each X_i is $Beta(\alpha_i, \alpha_0 - \alpha_i)$, its expectation is

$$E[X_i] = \frac{\alpha_i}{\alpha_0}$$

and its variance is

$$Var[X_i] = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)} \quad .$$

The Dirichlet distribution is the conjugate prior of the parameters of the multinomial distribution, meaning that if $(Y|x) \sim \mathrm{Multinomial}_q(n, x)$ and $X \sim Dir(\alpha)$ then $(X|Y = y) \sim Dir(\alpha + y)$. Importantly, the Dirichlet distribution can be used for inference of a multinomial distribution, since if the distribution is updated with samplings of a multinomial distribution ρ over outcomes $x_i, \ldots, x_q$ then each expectation $E[X_i]$ converges almost surely and in mean to $\rho(x_i)$.

3 Relative Vulnerability

This sections considers the entropy/vulnerability that occurs depending upon the distribution used by the attacker. The results in this section depend on the definition of vulnerability and min-entropy used in the one-try attack scenario, and do not necessarily generalize to other entropy measures and scenarios.

To simplify the notation we define the maximums $\overline{\mathcal{M}}^{\rho}$ and minimums $\underline{\mathcal{M}}^{\rho}$ as sets of elements of another set $\mathcal{M}$ for a given probability distribution ρ (with domain $\mathcal{M}$) as follows

$$\overline{\mathcal{M}}^{\rho} = \{m \in \mathcal{M} \,.\, \forall m' \in \mathcal{M} \,.\, \rho(m) \geq \rho(m')\}$$
$$\underline{\mathcal{M}}^{\rho} = \{m \in \mathcal{M} \,.\, \forall m' \in \mathcal{M} \,.\, \rho(m) \leq \rho(m')\} \,.$$

Note that each element in $\overline{\mathcal{M}}^{\rho}$ (resp. $\underline{\mathcal{M}}^{\rho}$) has the same probability; we will denote this probability as $\rho(\overline{m}^{\rho})$ (resp. $\rho(\underline{m}^{\rho})$).

Given two probability distributions ρ_G and ρ_A on the same domain we can consider a metric for their distance that aligns with min-entropy guessing attackers. Here ρ_G can be thought of as a generator for the elements $m \in \mathcal{M}$

or the true probability. Thus ρ_A can be considered the probability according to the attacker. Intuitively this is the vulnerability if the attacker's probability ρ_A is used to guess when ρ_G is the actual probability distribution. Let $\overline{\mathcal{M}}^{\rho_A}$ be partitioned into the sets $\overline{\mathcal{M}}^{\setminus} = \overline{\mathcal{M}}^{\rho_A} \setminus \overline{\mathcal{M}}^{\rho_G}$ of the wrong guesses of A and $\overline{\mathcal{M}}^{\cap} = \overline{\mathcal{M}}^{\rho_A} \cap \overline{\mathcal{M}}^{\rho_G}$ of the right guesses of A.

Note that the vulnerability of the message is written $V(M)$ implicitly assuming that there is a single probability distribution $\rho(M)$ over $\mathcal{M}$. In this section we are considering two different distributions $\rho_G(M)$ and $\rho_A(M)$ on the same set of messages, so we will write the corresponding vulnerabilities $V(G) \doteq V(\rho_G(M))$ and $V(A) \doteq V(\rho_A(M))$ for readability.

Definition 7. *The* relative vulnerability of ρ_G using ρ_A $\delta(G \to A)$ *is defined as follows*

$$\delta(G \to A) = \sum_{m \in \overline{\mathcal{M}}^{\setminus}} \frac{\rho_G(\overline{m}^{\rho_G}) - \rho_G(m)}{|\overline{\mathcal{M}}^{\rho_A}|} .$$

The definition above implicitly assumes that when the attacker has multiple possible max-probability outcomes to choose from, they will choose one of them at random with uniform probability. This is justified by the fact that, since such outcomes have the same probability in ρ_A, the attacker has no information to prefer any of them over the others, so he chooses one of them at random following the Maximum Entropy Principle.

Some Properties of $\delta(G \to A)$

This section formalizes some properties of relative vulnerability.

Lemma 1. $0 \leq \delta(G \to A) \leq \rho_G(\overline{m}^{\rho_G}) - \rho_G(\underline{m}^{\rho_G})$ *with*

- *equality to the left iff* $\overline{\mathcal{M}}^{\setminus} = \emptyset$*; and*
- *equality to the right iff* $\forall m \in \overline{\mathcal{M}}^{\rho_A}.\rho_G(m) = \rho_G(\underline{m}^{\rho_G})$.

The following corollaries immediately follow:

Corollary 1. *If* ρ_G *is the uniform distribution then* $\delta(G \to A)$ *for all* ρ_A *is 0.*

Corollary 2. $0 \leq \delta(G \to A) \leq 1$.

Importantly, relative vulnerability has the property of being equivalent to the loss of vulnerability when using the distribution ρ_A to guess the secret of a system that is generated with distribution ρ_G. Let's define the vulnerability $V(G \to A)$ of a system with distribution ρ_G when the attacker uses distribution ρ_A:

Definition 8. *The vulnerability* $V(G \to A)$ *of* ρ_G *using* ρ_A *is*

$$\frac{1}{|\overline{\mathcal{M}}^{\rho_A}|} \left(|\overline{\mathcal{M}}^{\cap}| \rho_G(\overline{m}^{\rho_G}) + \sum_{m \in \overline{\mathcal{M}}^{\setminus}} \rho_G(m) \right) .$$

Lemma 2. $V(G) - \delta(G \to A) = V(G \to A)$

The concept of relative vulnerability can be easily extended to conditional probabilities and vulnerabilities:

Definition 9. *The* relative vulnerability $\delta(G \to A|X)$ of ρ_G using ρ_A given X *is defined as follows*

$$\delta(G \to A|X) = \sum_{x \in \mathcal{X}} \rho_G(x)\delta(G|x \to A|x) \ .$$

Similarly we can define the vulnerability $V(G \to A)$ of a secret with distribution ρ_G when the attacker uses distribution ρ_A after observing X:

Definition 10. *The vulnerability* $V(G \to A|X)$ *of* G *using* A *after observing* X *is*

$$V(G \to A|X) = \sum_{x \in \mathcal{X}} \rho_G(x)V(G|x \to A|x)$$

and the following lemma holds:

Lemma 3. $V(G|X) - \delta(G \to A|X) = V(G \to A|X)$

We can use the results in this section to prove that if an attacker tries to guess a message generated by a distribution ρ_G while instead using distribution ρ_A after observing a variable X, the resulting min-entropy equivocation $H_\infty(G \to A|X)$ is greater than or equal to the min-entropy equivocation $H_\infty(G|X)$ obtained by using the distribution ρ_G, with equality iff $\delta(G \to A|X) = 0$.

Theorem 1.

$$H_\infty(G \to A|X) \geq H_\infty(G|X)$$

with equality iff $\delta(G \to A|X) = 0$.

Note that extending Theorem 1 to scenarios other than one-try attacks requires reasoning about the appropriate definitions of vulnerability. We refer to Köpf and Basin [17] for discussion of the connection between scenarios and uncertainty measures.

4 Encoder Rank

This section considers the information the attacker can gain merely from the encoder function a priori.

Observe that from the encoder function the message space $\mathcal{M}$ and the key space $\mathcal{K}$ can be inferred. Further, some properties of the message distribution can also be inferred from the encoder function. For example, consider the encoder functions for $|\mathcal{M}| = 3$ and $|\mathcal{K}| = 2$ in Table 1, reprinted from [5].

The encoder function in Table 1a induces the constraints $\rho(m_1) = 1 - 2\rho(c_3)$ and $\rho(m_2) = 1 - 2\rho(c_1)$. Similarly, Table 1b induces $\rho(m_1) = \rho(c_1) + \rho(c_4)$

Table 1: Encoder functions for $|\mathcal{M}| = 3$ and $|\mathcal{K}| = 2$.

		Message		
		m_1	m_2	m_3
Key	k_1	c_1	c_2	c_3
	k_2	c_2	c_3	c_1

(a) A Latin rectangle encoder

		Message		
		m_1	m_2	m_3
Key	k_1	c_1	c_2	c_3
	k_2	c_4	c_3	c_2

(b) A non-Latin rectangle encoder

and $\rho(m_2) + \rho(m_3) = \rho(c_2) + \rho(c_3)$. Naturally, in both cases we know that $\rho(m_1)+\rho(m_2)+\rho(m_3) = 1$ since ρ is a probability distribution over the messages.

We now introduce a convenient form to extract the constraints from the encoder function.

Definition 11. *The* ciphertext matrix form *of an encoder function is a* $|\mathcal{C}|\times|\mathcal{M}|$ *matrix* $\mathbf{C}$ *defined as*

$$\mathbf{C}(i,j) = \rho(c_i|m_j) = \sum_{k\in\mathcal{K}} \rho(c_i, k|m_j)$$

Fig. 1: Ciphertext matrix forms for the encoder functions in Table 1

$$\mathbf{C} = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \end{array} \begin{array}{c} \begin{array}{ccc} m_1 & m_2 & m_3 \end{array} \\ \begin{pmatrix} \rho(k_1) & 0 & \rho(k_2) \\ \rho(k_2) & \rho(k_1) & 0 \\ 0 & \rho(k_2) & \rho(k_1) \end{pmatrix} \end{array}$$

(a) Ciphertext matrix forms for the encoder function in Table 1a

$$\mathbf{C} = \begin{array}{c} \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{array} \begin{array}{c} \begin{array}{ccc} m_1 & m_2 & m_3 \end{array} \\ \begin{pmatrix} \rho(k_1) & 0 & 0 \\ 0 & \rho(k_1) & \rho(k_2) \\ 0 & \rho(k_2) & \rho(k_1) \\ \rho(k_2) & 0 & 0 \end{pmatrix} \end{array}$$

(b) Ciphertext matrix forms for the encoder function in Table 1b

The ciphertext matrix forms for the encoder functions in Table 1a and 1b are depicted in Figure 1a and 1b, respectively.

The ciphertext matrix form $\mathbf{C}$ can be seen as the coefficient matrix of a system of equations. Each row i in the augmented matrix represents the equation

$$\rho(c_i) = \rho(c_i|m_1)\rho(m_1) + \cdots + \rho(c_i|m_{|\mathcal{M}|})\rho(m_{\mathcal{M}})$$

where the $\rho(c_i|m_j)$ are the coefficients, the $\rho(m_j)$ are the variables, and $\rho(c_i)$ is the constant. Thus each row i corresponds to the equation $\rho(c_i) = \sum_{m\in\mathcal{M}} \rho(c_i, m)$. The equation $\sum_{m\in\mathcal{M}} \rho(m) = 1$ can be added to the system to represent the probability distribution constraint.

Since $\mathbf{C}$ is derived directly from the encoder function and we are assuming that the attacker knows the encoder function, then the attacker can compute $\mathbf{C}$ before observing any ciphertexts.

The number of solutions of the system depend on the rank $\texttt{rank}(\mathbf{C})$ of $\mathbf{C}$. We will refer to the rank of the ciphertext matrix form of an encoder function as *rank of the encoder* for brevity. We will show that the rank is an important property of the encoder since it defines the number of degrees of freedom of the message probability distribution according to the attacker. Consequently, among two encoder functions achieving similar equivocation, the one with the smallest rank should be preferred, since it will be more effective in hiding the message distribution from the attacker.

Recall by the Rouché-Capelli theorem that the number of solutions of the system of constraints has $|\mathcal{M}| - \texttt{rank}(\mathbf{C})$ degrees of freedom. The following theorem is an immediate consequence.

Theorem 2. *The number of possible message distributions inferred by the attacker from the observation of ciphertexts is an infinitude with* $|\mathcal{M}| - \texttt{rank}(\mathbf{C})$ *degrees of freedom.*

Corollary 3. *If* $|\mathcal{M}| = \texttt{rank}(\mathbf{C})$ *then the attacker can infer the exact message distribution from the observation of ciphertexts.*

Note that in general the rank of the matrix depends on the key distribution. Given a fixed distribution on the key space, the system of equations has at least one solution, i.e. where the probabilities of the messages correspond to the generator's distribution. Consequently, the rank of the coefficient matrix and the rank of the augmented matrix coincide by the Rouché-Capelli theorem.

5 Inferring Distributions

This section considers the capability of the attacker to infer the message or key distributions by observing the ciphertexts. The constraints from the encoder function can be combined with the ciphertext distribution to produce the augmented matrix. The attacker can use this information to converge upon a message distribution. Of course if the matrix has any degrees of freedom then there are infinite possible message distributions that can satisfy the constraints, and the attacker can only converge to one of these. The rest of this section builds up to these results.

For this section we assume that the attacker has access to the ciphertexts as well as the encoder function.

The base case to consider is when the attacker has access to the messages themselves. We show that in this case the attacker can determine the generator's probability distribution over the message space.

We take that $X = \{X_1, \ldots, X_{|\mathcal{M}|}\}$ is a vector of random variables where each random variable X_i for $1 \leq i \leq |\mathcal{M}| - 1$ represents the probability of the message m_i and $X_{|\mathcal{M}|} = 1 - \sum_{i=1}^{|\mathcal{M}|-1} X_i$. Let o_i be the number of times

that message m_i has been observed by the attacker and o the total number of messages observed. Then the attacker can use a Dirichlet distribution to infer the message distribution.

Theorem 3. *Let the attacker's posterior probability distribution ρ_A over the messages be*

$$\rho_A(m_i) = \frac{o_i + \rho_{AI}(m_i)}{o+1}$$

where ρ_{AI} is the attacker's prior. Then ρ_A tends to ρ_G as $o \to \infty$.

This allows the attacker to converge on the generator's distribution by observing messages. Observe that the initial state before any observations yields that $\rho_A(m_i) = \frac{\rho_{AI}(m_i)}{1}$ which represents the attacker's prior knowledge of the distribution.

Now consider the case in which the attacker does not have access to the messages, but only to the ciphertexts and the encoder function. We assume that the key distribution is uniform, and further that the attacker is aware of this.

It is straightforward to adapt Theorem 3 to consider the ciphertexts instead of messages. Here o_i is instead the number of times a ciphertext c_i has been observed by the attacker and o the total number of ciphertexts observed. Also let ρ_C be the actual probability distribution over the ciphertexts induced by ρ_G, the key distribution, and the encoder function. Similarly let ρ_{CI} be the prior ciphertext distribution calculated by the attacker before any observations have taken place. Again, the attacker uses a Dirichlet distribution to infer the ciphertext distribution.

Theorem 4. *Let the attacker's posterior probability distribution ρ_A over the ciphertexts be*

$$\rho_A(c_i) = \frac{o_i + \rho_{CI}(c_i)}{o+1} \quad .$$

Then ρ_A tends to ρ_C as $o \to \infty$.

The above result assumes the attacker does not adjust their prior after each observation, however it is straightforward to reproduce the results with this adaptation.

We now consider how the attacker can converge upon a message distribution by also exploiting the information gained from the encoder in the previous section.

From the encoder function and key distribution we can define a function $\textsc{Weight}(m, c)$ that given a message m and ciphertext c gives a weighted probability of the message m having been encoded to c. Thus define $\textsc{Weight}(m, c) = \sum_{k \in \mathcal{K}} \rho(k) \;.\; enc(m, k) = c$. Intuitively $\textsc{Weight}(m, c)$ is the likelihood considering of the message m to have been encoded to the ciphertext c.

Let w_i be the sum of the $\textsc{Weight}(m_i, c)$'s for all the observed ciphertexts c. This will now be used to converge upon a message distribution. Intuitively this is the same approach as used in Theorems 3 & 4 except now each observed

ciphertext is distributed over the possible messages that could have been used to produce it (and weighted by the key distribution).

This approach allows the attacker to use a Dirichlet distribution to converge on a distribution that satisfies the constraints of the encoder function and the generator's distribution (when the key space is less than the message space).

Theorem 5. *Let the attacker's posterior probability distribution ρ_A over the messages be*

$$\rho_A(m_i) = \frac{w_i + \rho_{AI}(m_i)}{o+1}$$

where ρ_{AI} is the attacker's prior. When $|\mathcal{K}| < |\mathcal{M}|$ then ρ_A tends to ρ'_A as $o \to \infty$ such that ρ'_A is a solution to the constraints induced by the encoder function.

Corollary 4. *Let the attacker's posterior probability distribution ρ_A over the messages be*

$$\rho_A(m_i) = \frac{w_i + \rho_{AI}(m_i)}{o+1}$$

where ρ_{AI} is the attacker's prior. When $|\mathcal{K}| < |\mathcal{M}|$ and $|\mathcal{M}| = \mathtt{rank}(\mathbf{C})$ then ρ_A tends to ρ_G as $o \to \infty$.

The above results rely upon the assumption that the key space is strictly less than the message space, otherwise it is possible for the attacker's distribution to remain unchanged or even be made inaccurate. By taking $|\mathcal{K}| = |\mathcal{M}|$ and using a Latin encoder with uniform key distribution then the rank of the encoder becomes 1 and so for any $|\mathcal{K}| > 1$ there are an infinite of solutions, indeed this is perfect secrecy or max-equivocation. For $|\mathcal{K}| > |\mathcal{M}|$ consider when $|\mathcal{K}| = |\mathcal{M}|+1$; then every ciphertext could have at least two keys mapping back to a particular message m. This would eventually lead to more weight being given to m than any other message, and converging on m being more likely than all others regardless of other information.

All the results above focus upon the attacker inferring knowledge of the message distribution, however they can be rephrased to instead infer information about the key distribution by swapping the rôles of key and message in the results.

The above results show that if an encoder function has rank equal to the message space then the attacker can uniquely determine the message distribution.

Consider again the encoder functions in Table 1 and the corresponding ciphertext matrices in Figure 1 and assume that the keys and messages are uniformly distributed. Then both encoder functions achieve max-equivocation for one-try attack scenarios. However, the ciphertext matrix in Figure 1a has rank 3 and 3 possible messages, thus $3 - 3 = 0$ degrees of freedom, meaning that the attacker can infer exactly the message distribution from the ciphertext distribution. On the other hand the ciphertext matrix in Figure1b has rank 2, meaning that the attacker will reduce the possible message distributions to an infinitude with a

single degree of freedom. While in this example one of the encoder is strictly better than the other, often a trade-off between equivocation and degrees of freedom is necessary. When the sender constructs the encoder function, they have to decide whether a loss of equivocation in favor of an increase in degrees of freedom is acceptable, evaluating what they know about the attacker's knowledge of the message distribution.

6 Curried Decoder Functions

In Sections 4 & 5 we have shown how knowledge of the encoder function can be used by the attacker to infer information about the message distribution, up to the number of degrees of freedom of the encoder function. Consequently, this section considers defenses against this information leakage by the sender not making an encoder function publicly known a priori. This can be easiest understood by considering the scenario where the sender wishes to send exactly one message to the receiver.

We consider when an encoder function is not fixed a priori and thus the sender can produce an encoder function that is unknown to the attacker and receiver. This implies that the sender communicates the produced encoder (or equivalently, the decoder) function to the receiver. Consequently, the attacker is assumed to intercept any communication of this encoder function, and so the attacker and receiver have the same information about the encoder function.

A trivial approach is for the sender to transmit the entire encoder function to the receiver. This solves the problem of decoding for the receiver, but discloses the entire encoder to the attacker and reduces to the same information leakage as in previous sections.

More interesting is to consider the advantages that can be gained by the sender constructing the encoder function and being able to choose how much of it to send.

One example of such advantages is to consider the possible encoders for a message space $\mathcal{M} = \{m_1, m_2, m_3\}$ and key space $\mathcal{K} = \{k_1, k_2\}$. In particular consider when $\rho_G(m_1) = 0.02$ and $\rho_G(m_2) = 0.49$ and $\rho_G(m_3) = 0.49$ and the encoder given in Table 1b. This encoder achieves the highest possible min-entropy equivocation and also has one degree of freedom, making it the best possible encoder for this message distribution. However, if the sender wishes to send the message m_1 then the leakage is total since all possible ciphertexts reveal the message (and the key). This occurs because since m_1 has only a 2% probability of being transmitted, it is convenient to optimize the encoder for the other 98% (that achieve max-equivocation). When only the message m_1 is to be transmitted, then the encoder in Table 1a achieves better equivocation for m_1, although it would be worse for m_2 or m_3.

The easiest scenario to consider is when the sender wishes to only send a single message m from a message space $\mathcal{M}$ with associated generator's distribution ρ_G (with domain $\mathcal{M}$). Clearly the sender has to transmit to the receiver sufficient information to reconstruct the message from the shared key k from key space

$\mathcal{K}$. This can be done with a *curried decoder* function $d : \mathcal{K} \to \mathcal{M}$ such that $d(k) = m$.

Consider the set $\mathcal{D}$ of all possible curried decoder functions, i.e. all functions from $\mathcal{K}$ to $\mathcal{M}$. Assume a function $pick : \mathcal{M} \times Dist(\mathcal{M}) \to Dist(\mathcal{D})$ producing a probability distribution $\rho(d|m)$ over each curried decoder $d \in \mathcal{D}$ such that $d(k) \neq m \Rightarrow \rho(d|m) = 0$. Let D be a random variable representing the choice of the decoder function. The sender then chooses a particular curried decoder $\bar{d}$ according to $\rho(D)$. The sender then transmits $\bar{d}$ to the receiver as its ciphertext, recall that $\bar{d}$ is also intercepted by the attacker as usual. Note that the probability distribution over D can be computed as $\forall d \in D.\ \rho(d) = \sum_{m \in \mathcal{M}} \rho(d|m)\rho_G(m)$.

Then we can immediately derive an upper bound for $H_\infty(M|D)$ corresponding to max-equivocation.

Lemma 4. $H_\infty(M|D) \leq H_\infty(K)$.

In this scenario of sending a single message m using key k it is straightforward to chose an encoder function that yields the ideal max-equivocation for $c = enc(m, k)$. This is possible since the encoder function need not be optimal (or even particularly good) for any other message as discussed above. Observe that when $|\mathcal{K}| = |\mathcal{M}|$ and the key distribution is uniform then this coincides with perfect secrecy and a one-time-pad.

To achieve the above results the naive *pick* function would map to a distribution assigning probability 1 to the curried decoder maximizing equivocation against an attacker knowing the generator's distribution ρ_G. This kind of worst-case assumption ensures that the attacker will not have a prior distribution closer to the real one than the one assumed by the sender. However, the choice of the curried decoder itself could reveal information about the message, which is why we wish the *pick* function to produce a distribution over the decoders. The following theorem computes the equivocation of the message depending on the choice of the decoder.

Theorem 6. *The expected message equivocation over all decoders can be computed as*

$$H_\infty(M|D) = -\log \sum_{d \in \mathcal{D}} \rho(d) \max_{m \in \mathcal{M}} \rho(m|d) \tag{1}$$

where $\rho(m|d) = \frac{\rho(d|m)\rho_G(m)}{\rho(d)}$.

The adaptation of Lemma 4 & Theorem 6 to entropy measures other than min-entropy is straightforward.

7 ExPad

In this section we present an algorithm to create a decoder d for a given message m achieving a good equivocation value by Theorem 6. That is, an algorithm that produces a good curried decoder function for a particular message without the choice of decoder revealing too much information about the message. Note that the algorithm produces only a curried decoder function, and does not create a complete encoder or decoder.

Data: message space $\mathcal{M}$, message to transmit $m \in \mathcal{M}$, probability distribution $\rho(M)$, key space $\mathcal{K}$, key k.
Result: curried decoder function d.

1 Let the ordered set $\mathbf{O}$ be the set $\mathcal{M}$ ordered by probability;
2 Let j be the position of m in $\mathbf{O}$;
3 Choose an index i uniformly between $\max(0, j - |\mathcal{K}| + 1)$ and $\min(|\mathbf{O}| - |\mathcal{K}|, j)$;
4 Let d be the ordered set obtained by enumerating $|\mathcal{K}|$ elements of $\mathbf{O}$ starting from index i included;
5 Randomize the order of the elements of d;
6 Switch the positions of element m and the element in position k in d;
7 Return d;

Algorithm 1: ExPad

7.1 The ExPad Algorithm

This section presents the ExPad algorithm and proves its effectiveness using the results in Section 6. Note that uniform distribution over the key is assumed for ExPad as presented here, although the algorithm can be adapted to account for non-uniform key distribution also.

The ExPad algorithm produces a curried decoder function $d : \mathcal{K} \to \mathcal{M}$, represented as an ordered list of $|\mathcal{K}|$ messages such that the message to be sent m is in position k of the list. Knowing the key k the receiver can immediately decode the message.

Now, order the elements of $\mathcal{M}$ according to their probability and denote this ordered set by $\mathcal{O}$. Let j be the position of m in $\mathcal{O}$. We create the decoder d by selecting a sequence of $|\mathcal{K}|$ elements from $\mathcal{O}$ including m. This ensures the probabilities of the elements of d are relatively close, since this yields the closest to uniform distribution and thus maximizes the equivocation. As explained in Section 6 we cannot choose d to maximize the equivocation since this may uniquely identify m. Instead ExPad exploits randomness here to make the choice of i. Thus, denote with j the index of m and choose randomly an index i to begin such that

$$max(0, j - |\mathcal{K}| + 1) \leq i \leq min(|\mathcal{M}|, j) \ .$$

Then d consists of the first $|\mathcal{K}|$ elements of $\mathcal{O}$ starting from i. These elements forming d are then randomly ordered into a sequence, with the element m fixed at position k. That is, the number corresponding to the message at the position of the value of the key. Now the curried decoder function d is this sequence. We call this algorithm ExPad and summarize it as Algorithm 1.

Observe that ExPad produces a curried decoder function for transmission of exactly one message. Extending to produce a sequence of curried decoder functions for a sequence of messages (with or without sharing the same keys), is a straightforward extension.

The next two sections consider an example of ExPad and then the message equivocation, respectively.

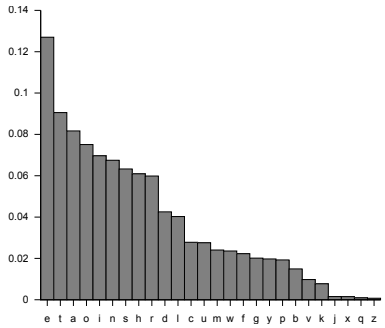

Fig. 2: Letters in the English language ordered by decreasing frequency.

7.2 ExPad Example

For a simple example, consider when the key is uniformly distributed over $\mathcal{K} = \{0, 1, 2, 3\}$ and the message is a single letter from some English text (`a`-`z`). Consider for simplicity the frequency of each letter in the English language [18] as the prior probability distribution $\rho(M)$, depicted in Fig. 2.

Assume that the sender wants to send the letter '`i`'. The index j of character '`i`' in the ordered distribution is 4. Then the sender chooses an index i from 1 to 4 with uniform probability and builds d with $|\mathcal{K}| = 4$ characters starting from i. Since we have a probability distribution over the characters, we can compute the entropy of each of the 4 possible ordered sets d. The possible sets d and their induced vulnerabilities and min-entropy equivocations are shown in Table 2.

Table 2: Curried decoder functions and their induced vulnerability and min-entropy equivocation for each index i in the ExPad example

	d	$V(M\|D=d)$	$H_\infty(M\|D=d)$
Index i	**1** `taoi`	0.2857	1.8073
	2 `aoin`	0.2778	1.8473
	3 `oins`	0.2724	1.8756
	4 `insh`	0.2665	1.9076

Assume that $i = 3$ and $k = 2$. Then the sender takes the ordered set $d = \{\texttt{o}, \texttt{i}, \texttt{n}, \texttt{s}\}$, puts the intended message '`i`' in position 2 and randomizes the order of the remaining messages, obtaining e.g. $\mathbf{C} = \{\texttt{o}, \texttt{n}, \texttt{i}, \texttt{s}\}$, and finally sends the whole set d to the receiver. Knowing that $k = 2$ the receiver identifies the message in position 2, i.e. '`i`', as the intended message.

7.3 Results for ExPad

The attacker intercepts the curried decoder function represented by d and tries to guess the message. Clearly the message is one of the $|\mathcal{K}|$ elements of d. Since the messages in d are ordered randomly, the probability of each of them being the original message is $1/|\mathcal{K}|$, thus the message equivocation of the decoder would be $H_\infty(M|D=d) = H_\infty(K)$. However, as explained in Section 6, the choice of the curried decoder itself reveals information about the message.

Let $\mathcal{M}_d$ be the set of messages for which decoder d could be chosen. In ExPad this corresponds to $|\mathcal{K}|$ elements of $\mathcal{M}$ that are adjacent in the ordered set $\mathbf{O}$. Then we can compute the expected equivocation for a decoder chosen by the ExPad algorithm.

Lemma 5. $\rho(m|d) = \frac{\rho_G(m)}{\sum_{m' \in \mathcal{M}_d} \rho_G(m')}$.

Recall from Theorem 6 that

$$H_\infty(M|D) = -\log \sum_{d \in \mathcal{D}} \rho(d) \max_{m \in \mathcal{M}} \rho(m|d)$$

thus if $\forall m \in \mathcal{M}, d \in \mathcal{D}.\ \rho(m|d) = 1/|\mathcal{K}|$ we would have $H_\infty(M|D) = H_\infty(K)$.

Since $\mathcal{M}_d$ is composed of $|\mathcal{K}|$ elements of $\mathcal{M}$ whose probability is close to m, then $\rho(m|d) = \frac{\rho_G(m)}{\sum_{m' \in \mathcal{M}_d} \rho_G(m')}$ is close to $1/|\mathcal{K}|$, and thus $H_\infty(M|D)$ is close to its upper bound of $H_\infty(K)$.

For example, the results of this for Table 2 yields an expected min-entropy equivocation of 1.8594, very close to 1.9076 bits of min-entropy equivocation achieved by the best encoder.

This holds for the example if the message is '`i`' because the probabilities around '`i`' are relatively smooth. For a character like '`d`', that is in a more diverse part of the probability distribution, the entropies of the possible curried decoder functions are lower; 1.6947 for ExPad with 1.8291 achieved by the best encoder.

Note that if $|\mathcal{M}| > 2|\mathcal{K}| - 2$ there will be $2|\mathcal{K}| - 2$ messages that can be mapped to less than $|\mathcal{K}|$ decoders, and will have a lower equivocation. However, it typically holds that $|\mathcal{M}| >> |\mathcal{K}|$ limiting the impact of this to a small number of messages.

In general, the equivocation will depend on how smooth the probability distribution over the ordered set $\mathbf{O}$ is. It has been shown [7] that it is possible to produce an encoder function achieving $H(M|C) = H(K)$ (or $H_\infty(M|C) = H_\infty(K)$) if and only if the message space $\mathcal{M}$ can be partitioned into subsets $\mathcal{M}_1, \ldots \mathcal{M}_k$ such that for each $1 \leq i \leq k$ it holds that $|\mathcal{M}_i| \geq |\mathcal{K}|$ and messages in $\mathcal{M}_i$ are equiprobable. Otherwise, as an example consider the distribution $\rho(b_i) = 1/2^i$ for a 4-valued key. The distribution is very skewed, so for any element the message equivocation is 0.9068 bits, not as close to the best encoders approaching 2 bits as for the English letter example.

8 SHORTPAD

While ExPad achieves a high equivocation, it requires transmission of a ciphertext linear in the size of the key space. In this section we present ShortPad, an encryption algorithm able to achieve similar equivocation to ExPad while transmitting a ciphertext logarithmic in the size of the key space.

8.1 The ShortPad Algorithm

The ShortPad algorithm assumes that the keys in $\mathcal{K}$ can be represented as bit strings of some uniform length, here denoted $|k|$, and that $|\mathcal{K}| = 2^{|k|}$. Further, also assume an operation $\odot$ that treats the elements of $\mathcal{M}$ as a group. The ShortPad algorithm exploits a pad consisting of an ordered set $\mathcal{R}$ of $|k|$ pad elements $r_1, \ldots, r_{|k|}$. For a key k, let $\mathcal{R}_k$ be a subset of $\mathcal{R}$ consisting exactly of the pad elements whose index corresponds to a 1 in the bit representation of k.

Define f to be the function that takes a seed element $s \in \mathcal{M}$ and a subset of elements of $\mathcal{M}$ and applies $\odot$ to all these elements (in order), i.e. $f(s, \{m_1, \ldots, m_i\}) = s \odot m_1 \odot \ldots \odot m_i$.

Observe that given a pad $\mathcal{R}$ and seed s the function f maps each $\mathcal{R}_k$ to some element of $\mathcal{M}$. Thus the choice of $\mathcal{R}$ and s induces the elements in the image of f. ShortPad exploits this by choosing $\mathcal{R}$ and s such that image of f for these yields good equivocation. Observe that the choice of $\mathcal{R}$ and s can be made in a similar manner to the choice of messages to send in ExPad. The remainder of ShortPad follows the same path, i.e. reducing to the $|\mathcal{K}|$ best possible pairs of $\mathcal{R}$ and s given m and then choosing one uniformly at random.

Aside, observe that many pads can be easily discarded from consideration due to relations within them that reduce equivocation. For example, if $r_i = r_j$ and $i \neq j$ this will trivially reduce the equivocation (since the two keys whose bit strings contain exactly one 1; at positions i and j will be equivalent) and so this pad can be discarded. Similar properties may hold for multiple r's depending upon the operation $\odot$.

Finally, the sender transmits the chosen pad $\mathcal{R}$ and seed s to the receiver allowing them to determine the correct message m (via using $f(\mathrm{s}, \mathcal{R}_k)$). As usual the attacker intercepts $\mathcal{R}$ and s.

Observe that the size of the pad is logarithmic in the size of the key space rather than linear as for ExPad, and the size of the seed is constant.

In general fixing a particular operation $\odot$ may have significant negative impact on the equivocation. This can be addressed by having several operations to choose from, and then specifying the chosen operation along with the pad and seed. This is a straightforward extension, that requires the transmission of a few extra bits to select from a potentially vast collection of operations, easily yielding good equivocation results.

8.2 ShortPad Example

Since ShortPad can be considered a compressed variant of ExPad, this section revisits the ExPad example from Section 7.2. Observe that$\mathcal{K} = \{0, 1, 2, 3\}$ can

be represented as bit strings by $\mathcal{K} = \{00, 01, 10, 11\}$, respectively. Again the message is a single letter from some English text (`a-z`) with prior probability as before. Let us assume that there is only one possible $\odot$ here that is the sum of the position of the letter in lexicographical order modulo 26.

Again assume that the sender wishes to send the letter '`i`' and the key is $2 = 10$. Now the four best possible pad and seed pairs are shown in Table 3 along with their vulnerability and min-entropy. Here the index i is used simply for the uniformly random choice of which pad and seed to use. Thus if the sender randomly chooses $i = 2$ then they would send pad '`og`' and seed '`u`'. Knowing that $k = 10$ the receiver recognises that the message is $s \odot r_1 = \mathtt{u} \odot \mathtt{o} = (20+14)\%26 = 8 = \mathtt{i}$.

Table 3: Pad and seed with their induced vulnerability and min-entropy equivocation in the ShortPad example

	i	$\mathcal{R}$	s	$V(M\|\mathcal{R}, s)$	$H_\infty(M\|\mathcal{R}, s)$
Index	**1**	`qb`	`s`	0.2905	1.7831
	2	`og`	`u`	0.3215	1.6368
	3	`pb`	`t`	0.3276	1.6096
	4	`qg`	`s`	0.3296	1.6010

8.3 Results for ShortPad

The only delicacy in the results is to manage the distribution of 1's and 0's in the bit string representations of the keys. This delicacy is elided by the condition that $|\mathcal{K}| = 2^{|k|}$. Then the results for ShortPad can be derived as straightforward adaptations of those for ExPad. The only other difference is in the manner of choosing the possible "ciphertexts" to send; i.e. sets of messages for ExPad; and pad, seed, and operator tuples for ShortPad. It is then straightforward to show that $\mathcal{R}$, s, and $\odot$ are equivalent to d.

8.4 Message-Independent ShortPad

The ShortPad algorithm above has exploited the combination of pad, seed, and operation to ensure particular values based upon the prior knowledge of the message to be sent. However, the majority of the algorithm can be used in a different manner that does not work only for a particular message. Observe that the pad and operation can in fact be used for any possible message by changing the choice of seed.

Recall the example in Section 8.2 where the message is '`i`'. This was obtained by the sender with $s \odot r_1 = \mathtt{u} \odot \mathtt{o} = (20 + 14)\%26 = 8 = \mathtt{i}$. However, any other message m' could be obtained simply by changing s such that $s \odot \mathtt{u} = m'$.

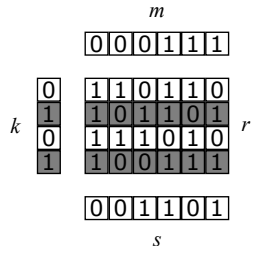

Fig. 3: Message-independent ShortPad example. The string s is produced as the exclusive disjunction of the message m with the rows of r corresponding to 1s in the key k (in grey).

This coverage of the whole message space means that the pad and operation can be used for any message. Indeed, the function $f(\cdot, \mathcal{R}_k)$ can be considered a decoder function with the seed s as the ciphertext. (The choice of pad and operation could also be part of the ciphertext as above, but for simplicity we consider them fixed here.) Thus by taking the inverse of this decoder function we achieve an encoder function that operates over the whole message space.

In the same manner that a number of operations can be fixed before hand and then indexed, a selection of pads and operations that yield good equivocation results for different subsets of the message space can be fixed in advance. This again reduces the information transmitted from logarithmic to a constant to determine the index of such a pad and operation pair.

This in some sense returns to the scenario where the encoders (now many possible encoders) are known to the attacker since the pad and operation combinations are already fixed. Thus, we could fix a single pad and operation to achieve the initial conditions of a known encoder, or balance the transmission cost of choosing from many possible encoders (i.e. pad and operation pairs) by sending the index of the chosen encoder. This essentially becomes a trade-off between pre-computing many possible encoders and looking up the best of the options, compared to computing the best possible encoder at transmission time. In either scenario the leakage can be quantified using the techniques presented in this paper.

Example Assume the following 4×6 pad r with degree 4 and no repeated columns, which has been pre-shared among all parties:

$$r = \begin{matrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{matrix} \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} .$$

Let $m = \texttt{000111}$ and $k = \texttt{0101}$. Since the second and fourth bits of the key are ones, we produce the bit string s as the exclusive disjunction $\oplus$ of the message

m with the second and fourth rows of r:

$$\begin{aligned} s &= m \oplus r_2 \oplus r_4 \\ &= \texttt{000111} \oplus \texttt{101101} \oplus \texttt{100111} \\ &= \texttt{001101} \end{aligned}$$

and we transmit s to the receiver. The example is also depicted in Figure 3.

9 Conclusions

We have considered several scenarios where an attacker attempts to gain information about a message or message distribution. We have shown that a one-try attack by an attacker with a different message distribution to the generator can only do worse than knowing the generator's distribution. This can be quantified as the relative vulnerability when using a different distribution.

We have shown that knowledge of the encoder function can yield information about the generator's message distribution. In particular, when the matrix representation of the encoder function has maximal rank, then there is a unique distribution that can be revealed with knowledge of the ciphertext distribution. Otherwise, when the rank is not maximal then each decrease in rank induces an infinitude of possible message distributions.

We show that the attacker can converge upon the ciphertext distribution by observing transmissions, when the encoder function is fixed regardless of the key(s) used. Combining this with the information from the encoder function, the attacker can converge on a message distribution that satisfies constraints induced by the encoder function. Further, when the matrix representation has maximal rank the attacker can converge upon the generator's distribution. Thus, when the attacker does not know the generator's distribution, maximizing equivocation alone may not be optimal since reduced matrix rank can prevent convergence to the generator's distribution.

We show that an encoder function need not be fixed, and revealed, in advance and so can be constructed by the sender as required. This allows for transmission of a curried decoder function that reveals less information about the distribution since much of the encoder function is never available to the attacker. Indeed, when sending only a small number of messages the encoder function cannot be fully available and so the matrix rank cannot be determined by the attacker. These curried decoder functions can be exploited to choose ones that maximize equivocation for the message(s) to be sent, optimizing for them rather than all possible messages. Although optimizing too strongly can leak information by indicating which message the optimization has been for.

The ExPad algorithm is presented as a straightforward approach to achieving high equivocation and sending a curried decoder function. We show that ExPad can achieve high equivocation for any particular message, not being limited like a traditional encoder function that must balance equivocation for all messages.

The ShortPad algorithm is presented as a way to reduce the data transmission volume required for ExPad while maintaining good equivocation properties. We show that the approach used in ShortPad could be exploited by pre-sharing several pads and choosing the best one when transmitting a message, reducing the transmission volume further.

Future Work

This paper makes several assumptions that can be relaxed to consider alternative scenarios, attacks, and defences.

Unknown Message Spaces We have assumed that the message space is perfectly known to all parties. However, there are scenarios where the message space may not be perfectly known to some parties. For example, the sender and receiver may have knowledge of their message space, while the attacker may be unaware or have some superset. In this case the attacker does not have a (straightforward) probability distribution over the messages.

If the sender and receiver can agree upon an indexing of the messages, then transmitting only the index prevents the attacker from recovering any information about the message. However, this becomes a security through obscurity scenario and any security guarantee upon it becomes completely dependent on the attacker not having information about the message space.

Infinite Message Spaces The message space here is considered finite, and although the results apply for any finite message space, in practice it may be more elegant to work over an infinite message space. This would allow results for transmission of extremely large files (e.g. Blu-ray movies) with good equivocation without needing to find encoders over all possible files. In this case we can assume that the attacker's knowledge is represented by a function mapping every finite subset of the message space to a probability distribution on that subset.

The sender can produce and send a curried decoder function as explained in Section 6, reducing the message space to the image of this curried decoder. Message equivocation can be computed as normal. While prior entropy and thus leakage would be convoluted to compute in this scenario, equivocation-based principles like the ones presented in this paper apply directly.

Stream Ciphers An alternative to considering such extreme message spaces would be to consider stream cipher variants of the results and algorithms presented here and in the related work. It may be that good equivocation can be achieved for stream ciphers that do not require reasoning over impractically large message spaces. The message-independent ShortPad variant can be easily modified to encode a stream, particularly when using a block-by-block operation like the exclusive disjunction example in Section 8.4.

Misleading the Attacker Another path of exploration is to consider how the cryptosystem may not merely defend against attacks, but actively mislead an attacker to gain false information. This is mentioned briefly as a possibility when large key spaces are available (in Section 5), but other techniques also show some promise. When the sender and receiver share some information (beyond the key) unknown to the attacker this could be exploited. For example, if there are invalid messages in the message space, these could be sent in place of valid messages, leading the attacker to believe they are valid and thus converge to an incorrect distribution over the message space.

References

1. P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. on Computing*, pp. 1484–1509, 1997.
2. F. L. Traversa, C. Ramella, F. Bonani, and M. Di Ventra, "Memcomputing NP-complete problems in polynomial time using polynomial resources and collective states," *Science Advances*, vol. 1, no. 6, 2015.
3. C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical Journal*, vol. 28, 1949.
4. ——, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, pp. 379–423, Jul. 1948.
5. F. Biondi, T. Given-Wilson, and A. Legay, "Attainable Unconditional Security for Shared-Key Cryptosystems," in *The 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (IEEE TrustCom-15)*, Helsinki, Finland, Aug. 2015. [Online]. Available: https://hal.inria.fr/hal-01192859
6. G. Smith, "On the foundations of quantitative information flow," in *FOSSACS 2009*, ser. LNCS, L. de Alfaro, Ed., vol. 5504. Springer, 2009, pp. 288–302.
7. F. Biondi, T. Given-Wilson, and A. Legay, "Attainable Unconditional Security for Shared-Key Cryptosystems," Nov. 2015, extended version; available at https://hal.inria.fr/hal-01233185. [Online]. Available: https://hal.inria.fr/hal-01233185
8. M. S. Alvim, K. Chatzikokolakis, C. Palamidessi, and G. Smith, "Measuring information leakage using generalized gain functions," in *CSF 2012*, S. Chong, Ed. IEEE, 2012, pp. 265–279.
9. A. McIver, C. Morgan, G. Smith, B. Espinoza, and L. Meinicke, "Abstract channels and their robust information-leakage ordering," in *Principles of Security and Trust - Third International Conference, POST 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014, Proceedings*, ser. Lecture Notes in Computer Science, M. Abadi and S. Kremer, Eds., vol. 8414. Springer, 2014, pp. 83–102. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-54792-8_5
10. M. Alimomeni and R. Safavi-Naini, "Guessing secrecy," in *Information Theoretic Security*, ser. Lecture Notes in Computer Science, A. Smith, Ed. Springer Berlin Heidelberg, 2012, vol. 7412, pp. 1–13. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-32284-6_1
11. M. S. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, C. Palamidessi, and G. Smith, "Additive and multiplicative notions of leakage, and their capacities," in *IEEE 27th Computer Security Foundations Symposium, CSF 2014, Vienna,*

Austria, 19-22 July, 2014. IEEE, 2014, pp. 308–322. [Online]. Available: http://dx.doi.org/10.1109/CSF.2014.29

12. C. Braun, K. Chatzikokolakis, and C. Palamidessi, "Quantitative notions of leakage for one-try attacks," *Electr. Notes Theor. Comput. Sci.*, vol. 249, pp. 75–91, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.entcs.2009.07.085
13. M. R. Clarkson, A. C. Myers, and F. B. Schneider, "Quantifying information flow with beliefs," *Journal of Computer Security*, vol. 17, no. 5, pp. 655–701, 2009. [Online]. Available: http://dx.doi.org/10.3233/JCS-2009-0353
14. S. Hamadou, V. Sassone, and C. Palamidessi, "Reconciling belief and vulnerability in information flow," in *31st IEEE Symposium on Security and Privacy, S&P 2010, 16-19 May 2010, Berleley/Oakland, California, USA*. IEEE Computer Society, 2010, pp. 79–92. [Online]. Available: http://dx.doi.org/10.1109/SP.2010.13
15. T. Cover and J. Thomas, *Elements of Information Theory*, ser. A Wiley-Interscience publication. Wiley, 2006.
16. S. Kotz, N. Balakrishnan, and N. Johnson, *Continuous Multivariate Distributions, Models and Applications*. Wiley, 2004.
17. B. Köpf and D. A. Basin, "An information-theoretic model for adaptive side-channel attacks," in *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007*, P. Ning, S. D. C. di Vimercati, and P. F. Syverson, Eds. ACM, 2007, pp. 286–296. [Online]. Available: http://doi.acm.org/10.1145/1315245.1315282
18. S. Mayzner, *Tables of Single-letter and Digram Frequency Counts for Various Word-length and Letter-position Combinations*, ser. Psychonomic monograph supplements. Psychonomic Press, 1965.

Omitted Proofs

The proofs of all non-trivial lemmata and theorems in the paper follow.

Omitted from Section 3

Proof (of Lemma 2).

$$
\begin{aligned}
&V(G) - \delta(G \to A) \\
&= \rho_G(\overline{m}^{\rho_G}) - \sum_{m \in \overline{\mathcal{M}}^{\setminus}} \frac{\rho_G(\overline{m}^{\rho_G}) - \rho_G(m)}{|\overline{\mathcal{M}}^{\rho_A}|} \\
&= \rho_G(\overline{m}^{\rho_G}) - \frac{1}{|\overline{\mathcal{M}}^{\rho_A}|} \left(|\overline{\mathcal{M}}^{\setminus}| \rho_G(\overline{m}^{\rho_G}) - \sum_{m \in \overline{\mathcal{M}}^{\setminus}} \rho_G(m) \right) \\
&= \rho_G(\overline{m}^{\rho_G}) - \frac{|\overline{\mathcal{M}}^{\setminus}|}{|\overline{\mathcal{M}}^{\rho_A}|} \rho_G(\overline{m}^{\rho_G}) + \frac{1}{|\overline{\mathcal{M}}^{\rho_A}|} \sum_{m \in \overline{\mathcal{M}}^{\setminus}} \rho_G(m) \\
&= \rho_G(\overline{m}^{\rho_G}) \left(1 - \frac{|\overline{\mathcal{M}}^{\setminus}|}{|\overline{\mathcal{M}}^{\rho_A}|} \right) + \frac{1}{|\overline{\mathcal{M}}^{\rho_A}|} \sum_{m \in \overline{\mathcal{M}}^{\setminus}} \rho_G(m) \\
&= \rho_G(\overline{m}^{\rho_G}) \left(\frac{|\overline{\mathcal{M}}^{\cap}|}{|\overline{\mathcal{M}}^{\rho_A}|} \right) + \frac{1}{|\overline{\mathcal{M}}^{\rho_A}|} \sum_{m \in \overline{\mathcal{M}}^{\setminus}} \rho_G(m) \\
&= \frac{1}{|\overline{\mathcal{M}}^{\rho_A}|} \left(|\overline{\mathcal{M}}^{\cap}| \rho_G(\overline{m}^{\rho_G}) + \sum_{m \in \overline{\mathcal{M}}^{\setminus}} \rho_G(m) \right) \\
&= V(G \to A) \, .
\end{aligned}
$$

Proof (of Lemma 3).

$$
\begin{aligned}
&V(G|X) - \delta(G \to A|X) \\
&= \sum_{x \in \mathcal{X}} \rho_G(x) V(G|x) - \sum_{x \in \mathcal{X}} \rho_G(x) \delta(G|x \to A|x) \\
&= \sum_{x \in \mathcal{X}} \rho_G(x) \left(V(G|x) - \delta(G|x \to A|x) \right) \\
&= \sum_{x \in \mathcal{X}} \rho_G(x) V(G|x \to A|x) \qquad \text{(by Lemma 2)} \\
&= V(G \to A|X) \, .
\end{aligned}
$$

Proof (of Theorem 1).

$$
\begin{aligned}
&H_\infty(G \to A|X) - H_\infty(G|X) \\
&= -\log V(G \to A|X) + \log V(G|X) \\
&= \log \frac{V(G|X)}{V(G \to A|X)} \\
&= \log \frac{V(G|X)}{V(G|X) - \delta(G \to A|X)} && \text{(by Lemma 3)}
\end{aligned}
$$

and the result follows.

Omitted from Section 4

Proof (of Theorem 2). Each degree of freedom allows an infinitude of message distributions, conclude by Rouché-Capelli.

Omitted from Section 5

Proof (of Theorem 3). The posterior probability distribution over X is a Dirichlet distribution $Dir(\alpha)$ where $\alpha = \{\alpha_1, \ldots, \alpha_{|\mathcal{M}|}\}$ and $\alpha_i = o_i + 1$. The expected value of X_i is $\frac{\alpha_i}{\alpha_0} = \frac{o_i + \rho_{AI}(m_i)}{o+1}$. Conclude by convergence of Dirichlet distribution to the sampled distribution.

Proof (of Theorem 4). The posterior probability distribution over X is a Dirichlet distribution $Dir(\alpha)$ where $\alpha = \{\alpha_1, \ldots, \alpha_{|\mathcal{M}|}\}$ and $\alpha_i = o_i + 1$. The expected value of X_i is $\frac{\alpha_i}{\alpha_0} = \frac{o_i + \rho_{CI}(c_i)}{o+1}$. Conclude by convergence of Dirichlet distribution to the sampled distribution.

Proof (of Theorem 5). The probability distribution over X is a Dirichlet distribution $Dir(\alpha)$ where $\alpha = \{\alpha_1, \ldots, \alpha_{|\mathcal{M}|}\}$ and $\alpha_i = d_i + 1$. The expected value of X_i is $\frac{\alpha_i}{\alpha_0} = \frac{d_i + \rho_{AI}(m_i)}{o+1}$. Conclude by convergence of Dirichlet distribution to the sampled distribution. That this satisfies the constraints of the encoder function is by construction.

Proof (of Corollary 4). By Rouché-Capelli there is exactly one solution and so this must coincide with that of the generator.

Omitted from Section 6

Proof (of Lemma 4). Note that since each $d \in \mathcal{D}$ is a function $d : \mathcal{K} \to \mathcal{M}$ then $H_\infty(M|D = d) \leq H_\infty(K)$ therefore $H_\infty(M|D) \leq H_\infty(K)$.

Proof (of Theorem 6). Trivial from the definition of conditional min-entropy.

Omitted from Section 7

Proof (of Lemma 5). Note that $m \notin \mathcal{M}_d \Rightarrow \rho(d|m) = 0$ by definition of the *pick* function. Therefore we compute $\rho(d)$ as

$$\begin{aligned}
\rho(d) &= \sum_{m \in \mathcal{M}} \rho(m, d) \\
&= \sum_{m \in \mathcal{M}_d} \rho(m, d) \\
&= \sum_{m \in \mathcal{M}_d} \rho_G(m) \rho(d|m) \\
&= \frac{1}{|\mathcal{K}|} \sum_{m \in \mathcal{M}_d} \rho_G(m)
\end{aligned}$$

and for each message m we compute $\rho(m|d)$ with Bayes' theorem as

$$\begin{aligned}
\rho(m|d) &= \frac{\rho(d|m)\rho_G(m)}{\rho(d)} \\
&= \frac{\frac{1}{|\mathcal{K}|}\rho_G(m)}{\frac{1}{|\mathcal{K}|}\sum_{m' \in \mathcal{M}_d} \rho_G(m')} \\
&= \frac{\rho_G(m)}{\sum_{m' \in \mathcal{M}_d} \rho_G(m')}.
\end{aligned}$$